Mapping and modeling Earth Science Data

Introduction & Segment I: UNIX

Thorsten Becker

University of Southern California, Los Angeles

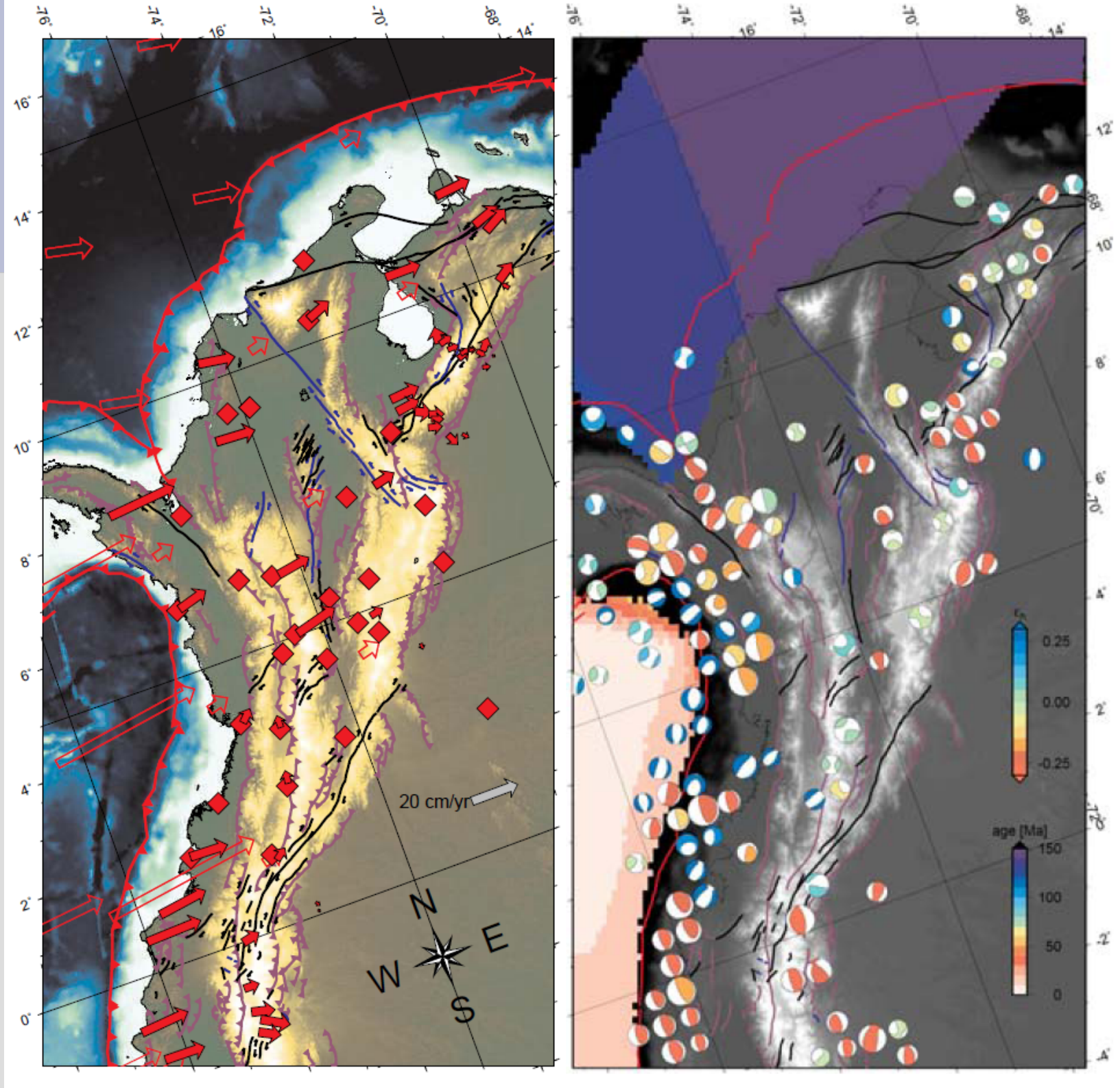
Università di Roma TRE, June 2012

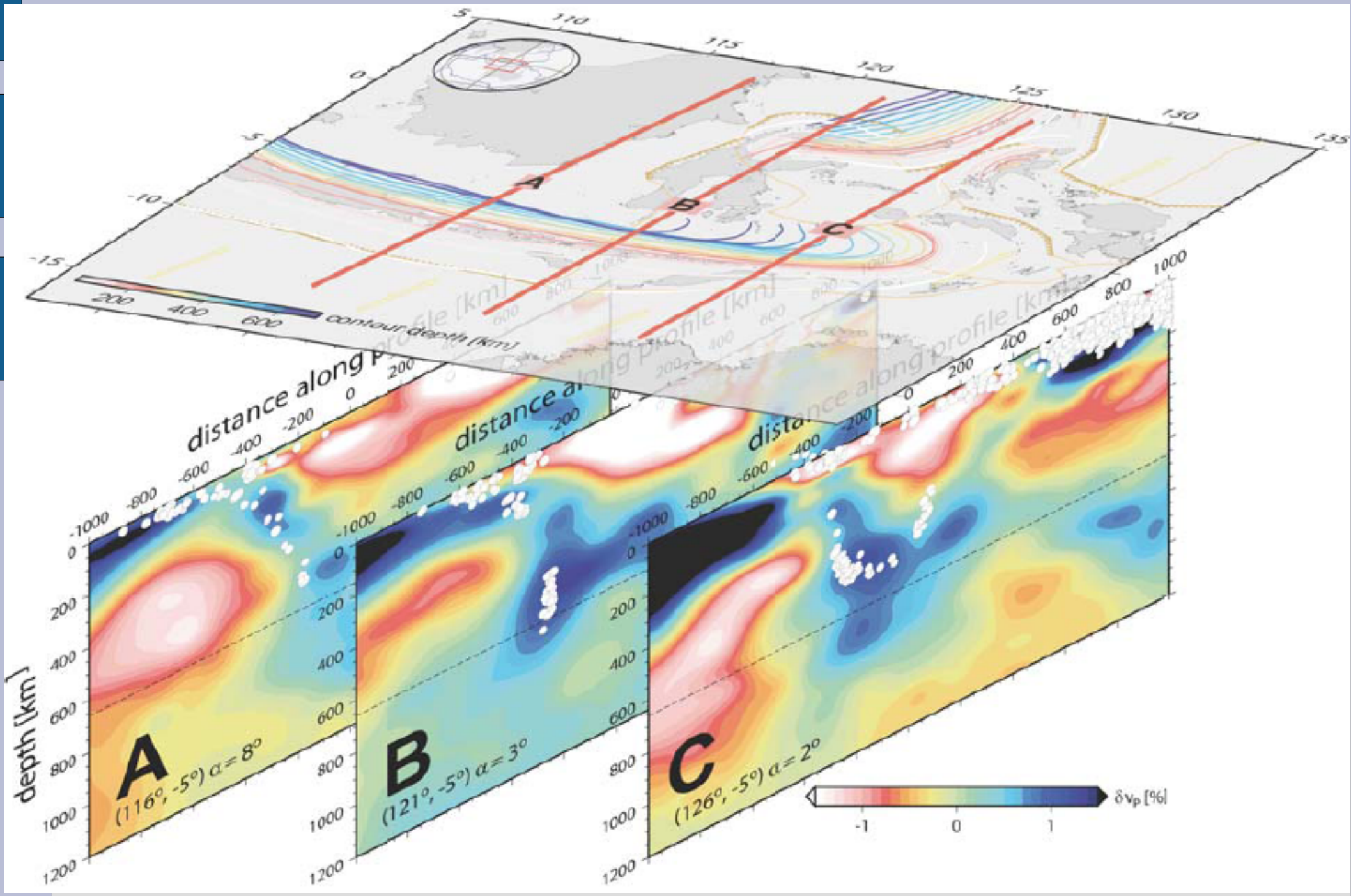
Introduction and scope of course

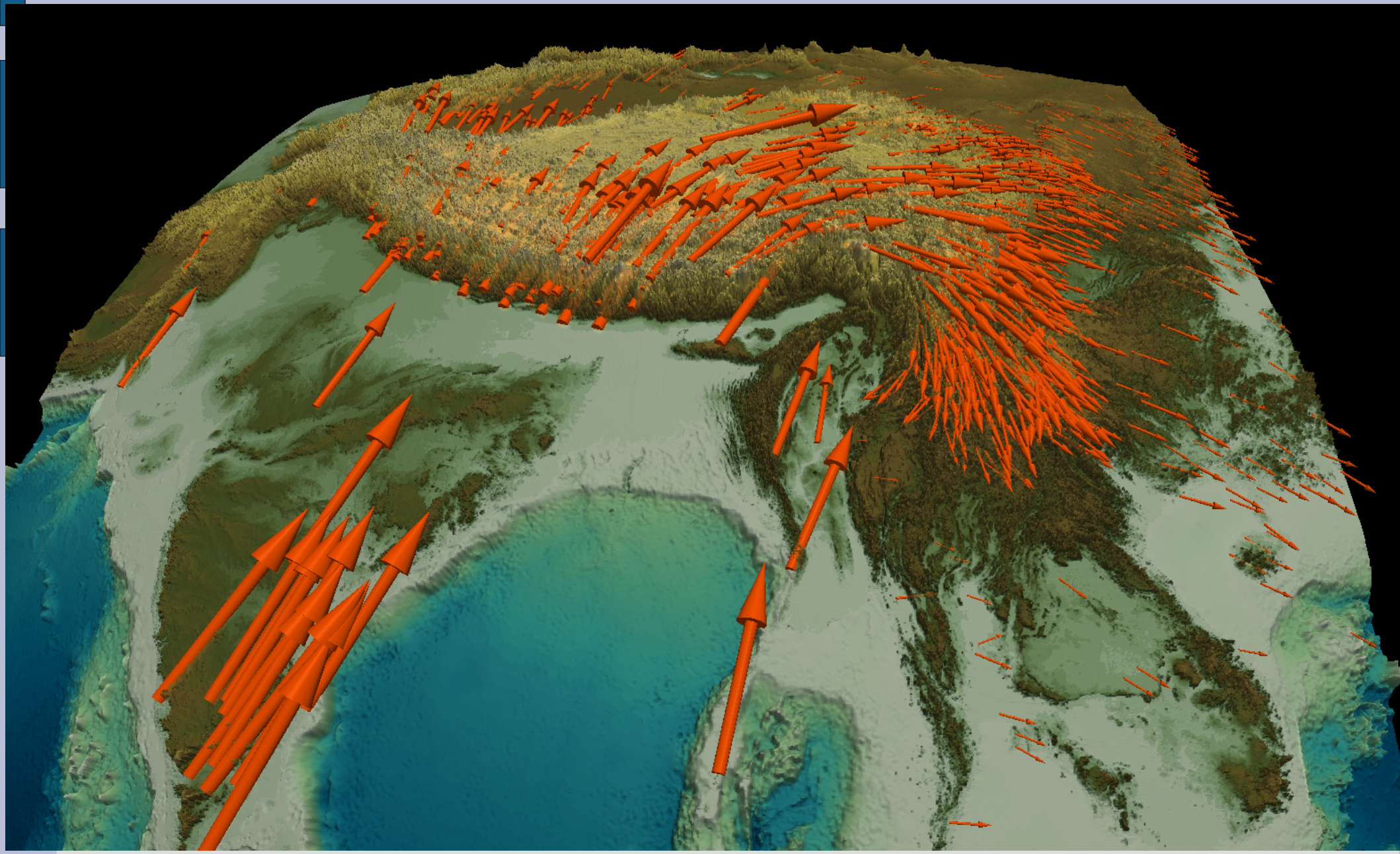
Who I am

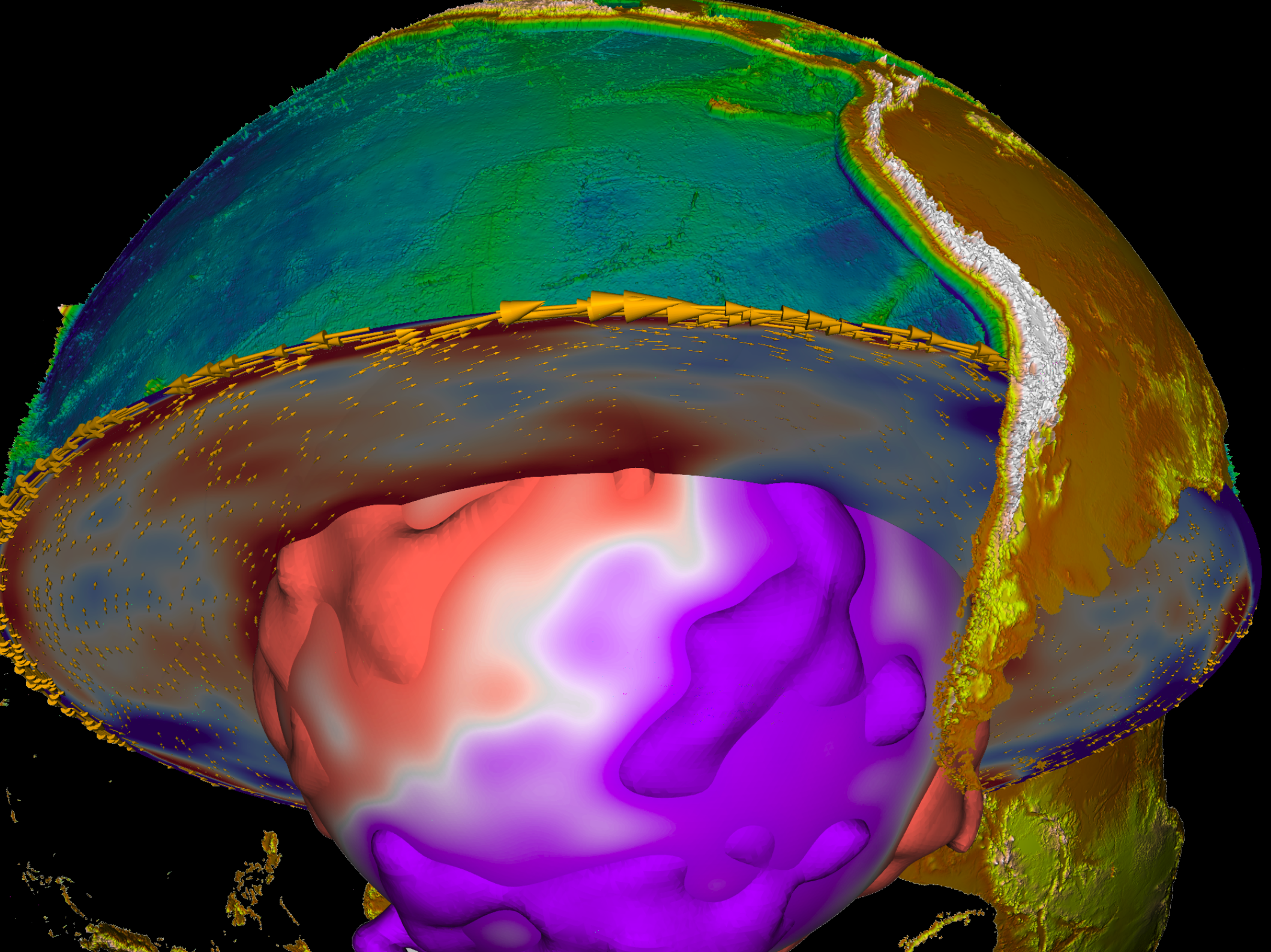
- Education: Geophysicist, trained as a physicist
- Objective: Solid Earth dynamics and evolution
- Disciplines: Geodynamics and seismology
- Tools: Numerical modeling

What I do

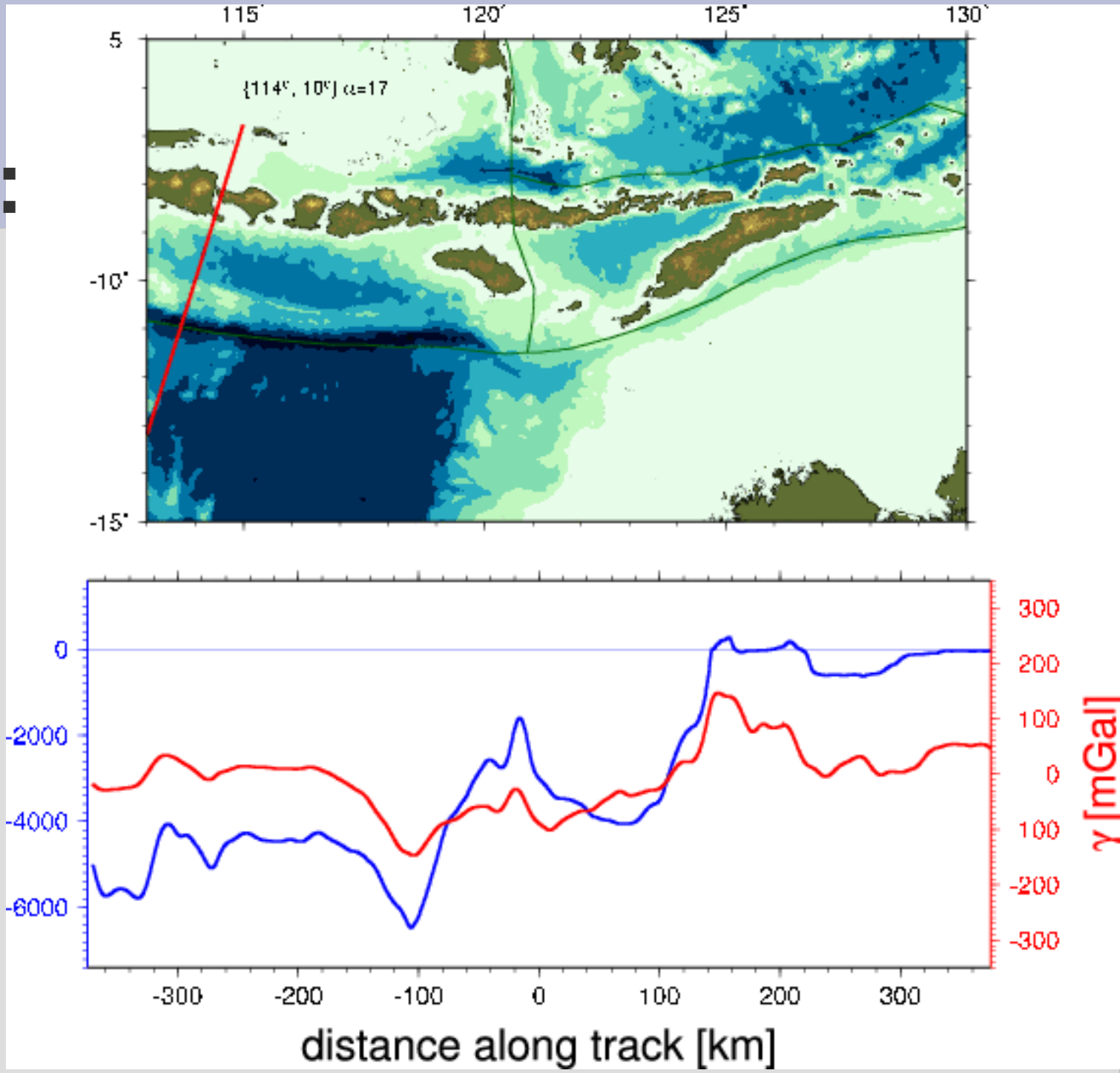



A
B
C
contour depth [km]
200
400
600
distance along profile [km]
depth [km]
(116°, -5°) α = 8°
(121°, -5°) α = 3°
(126°, -5°) α = 2°
δvp [%]
-1
0
1

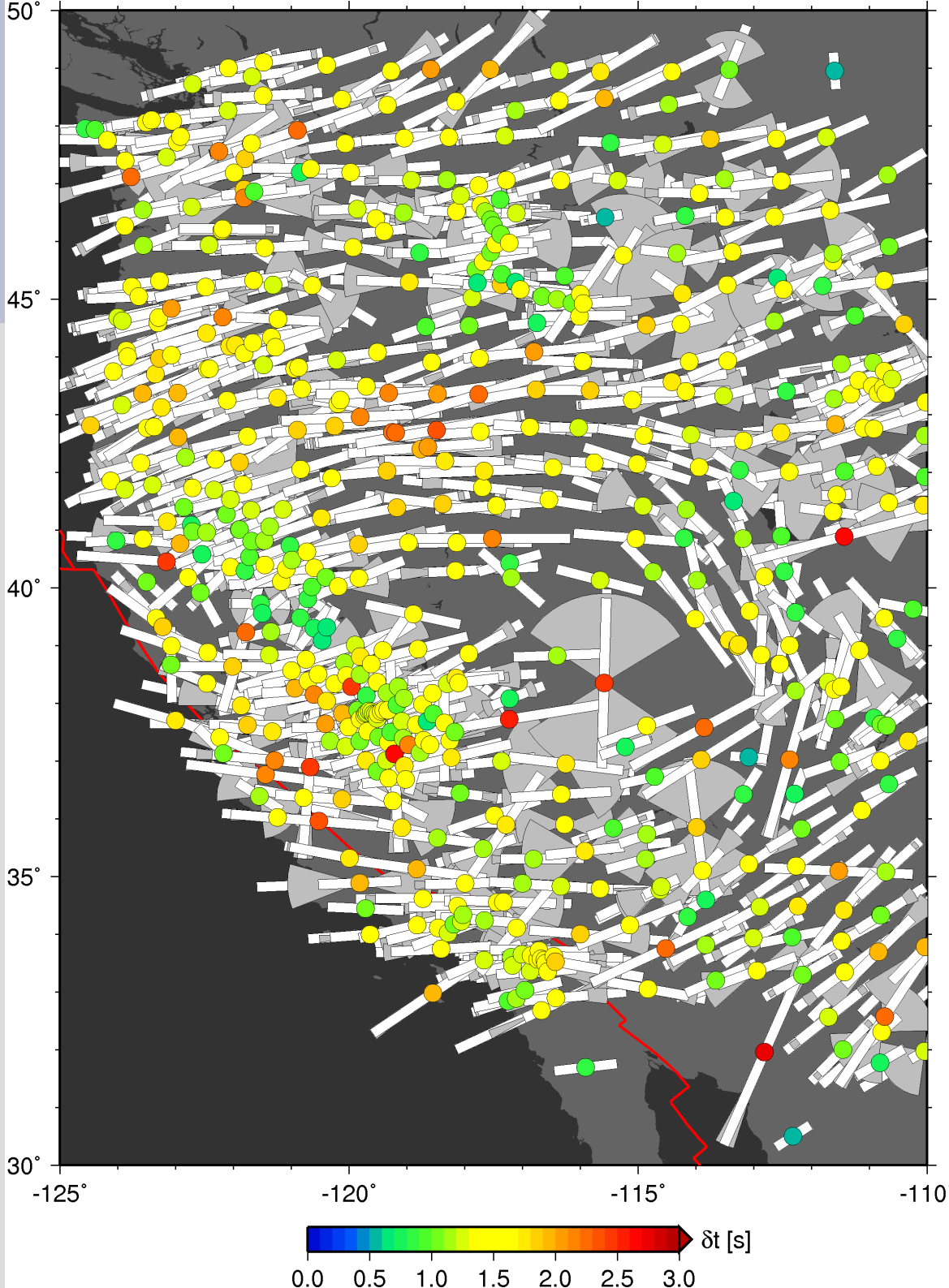




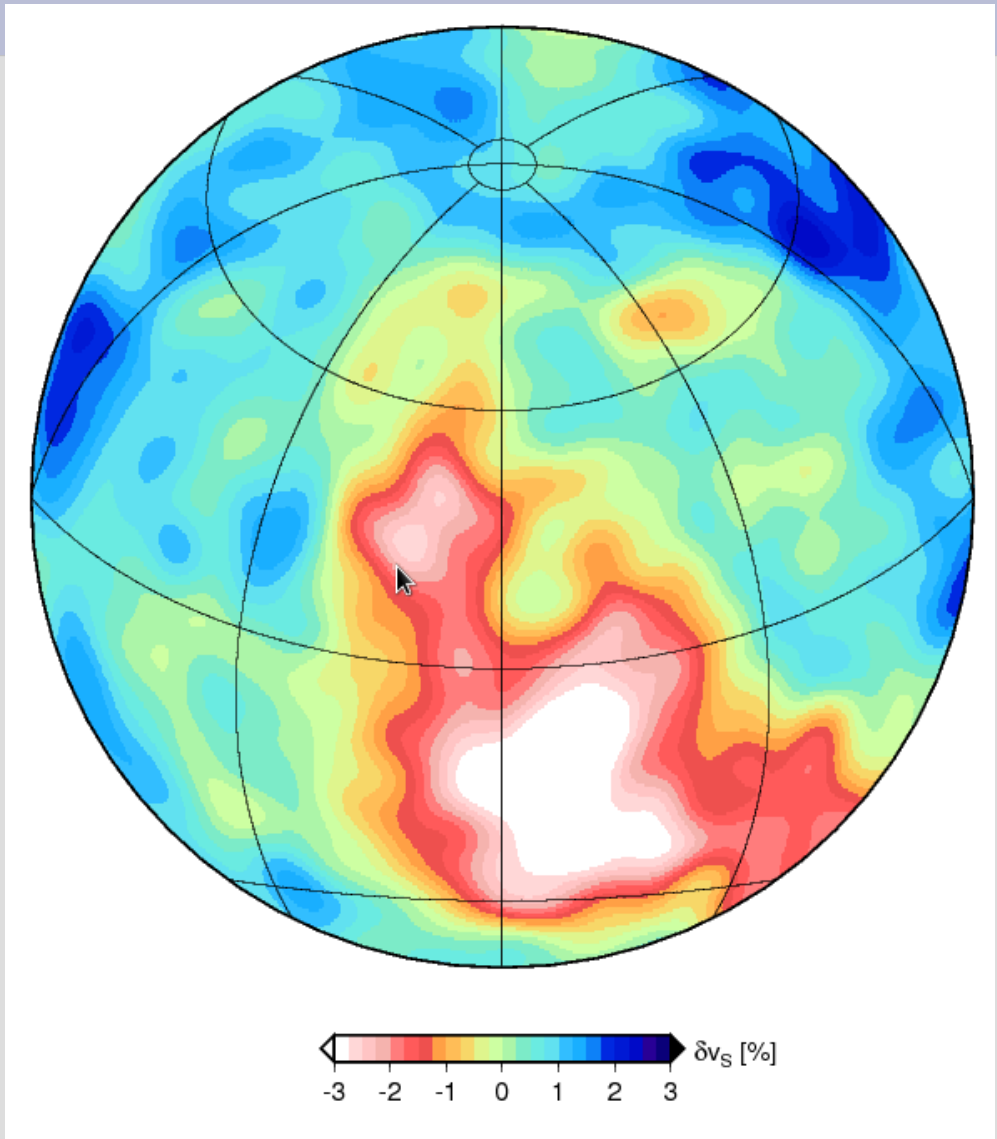
Scripting: Simple movies for data analysis

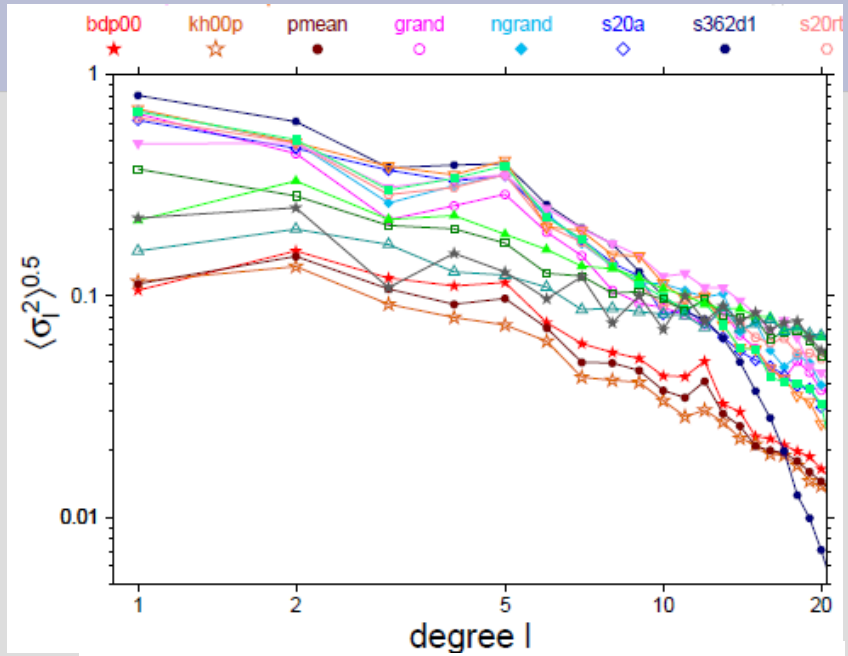


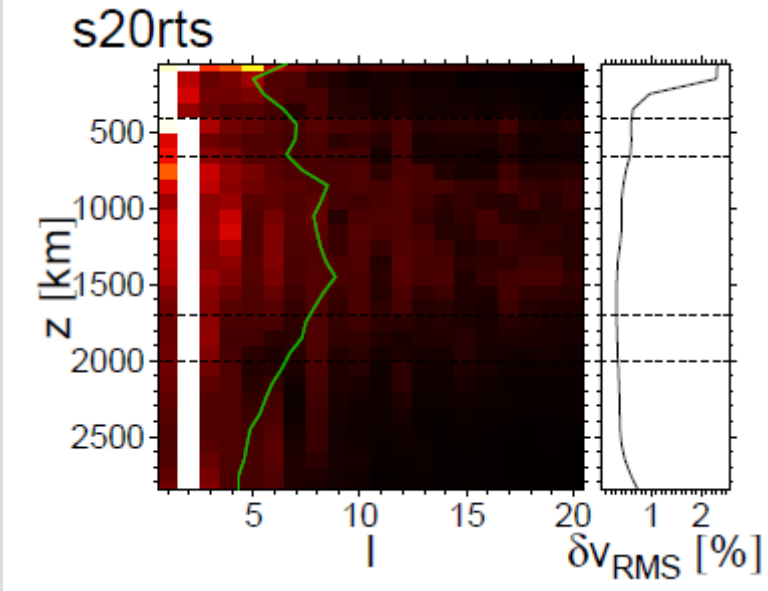
Scripting: Automated shear wave splitting



Data/model analysis







Purpose of this short course

- Introduce a scientific/numeric workflow setting centered around LINUX/UNIX
- Provide background for using the USC Earth Science Computing Environment
- **Provide skills and tools for making maps and analyzing Earth science data with GMT**
- Comment on some other analysis, visualization and typesetting tools
- *What is your objective/background?*

Contents

Operating system: UNIX (where we live)
Script programming (how we fiddle)

Mapping with GMT I (what we make)
Mapping with GMT II
Mapping with GMT III

Visualization and data analysis

Segment I: Computing and UNIX
Lecture 1

Purpose of this segment

- Introduce UNIX-based (e.g. LINUX, Mac OSX) computing and scientific work flow environments by providing pointers for further information
- Describe what I think are best practices in moderate to high-performance computing
  - I will make judgments and provide specific recommendations
  - I cannot possibly provide a comprehensive, fair, or entirely up to date overview

Typographic conventions

- Most important:
  - links to web based information are blue
- A lot of information (and examples) are on the web
- UNIX (or shell) commands are usually written in **bold**
- `Input into the shell is usually in Courier font.`

Contents segment I: UNIX

- UNIX (or LINUX, used synonymously here): what and why
- The file system and Window managers
- Shell environment
- Editing files
- Command line tools
- Scripts and GUIs

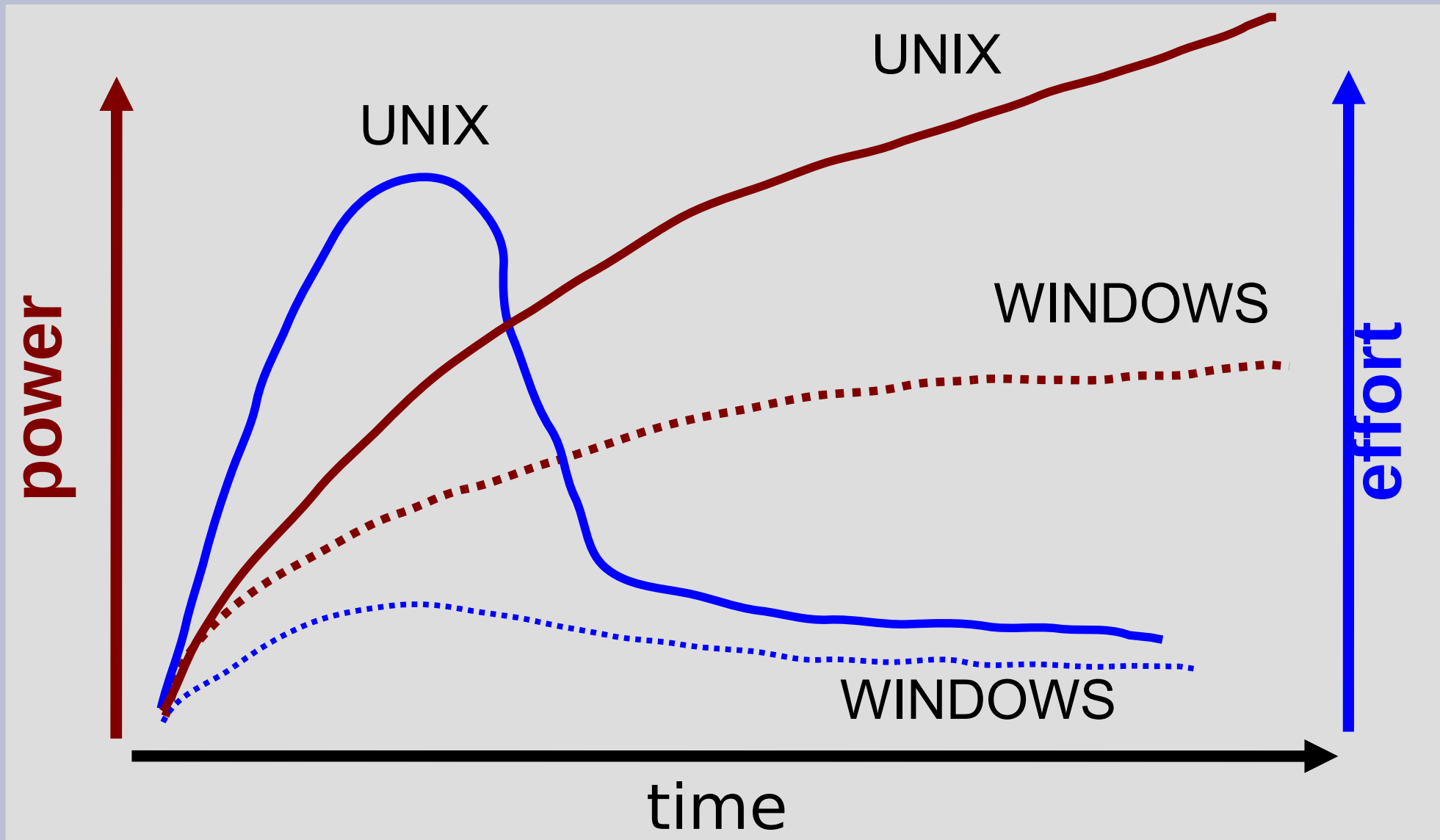
UNIX: What is UNIX?

- an operating system that originated in the 70ies
- build for multi-user, multi-tasking, scalable (that was new way back then)
- runs on all computing hardware, including iPOD
- many flavours, free: LINUX, BSD (a version made it into OSX), Solaris (SUN)
- they are all *kind of* the same thing, your mileage may vary (e.g. directory structure)
- there is convergence between LINUX, Mac OS, and Windows look and feel

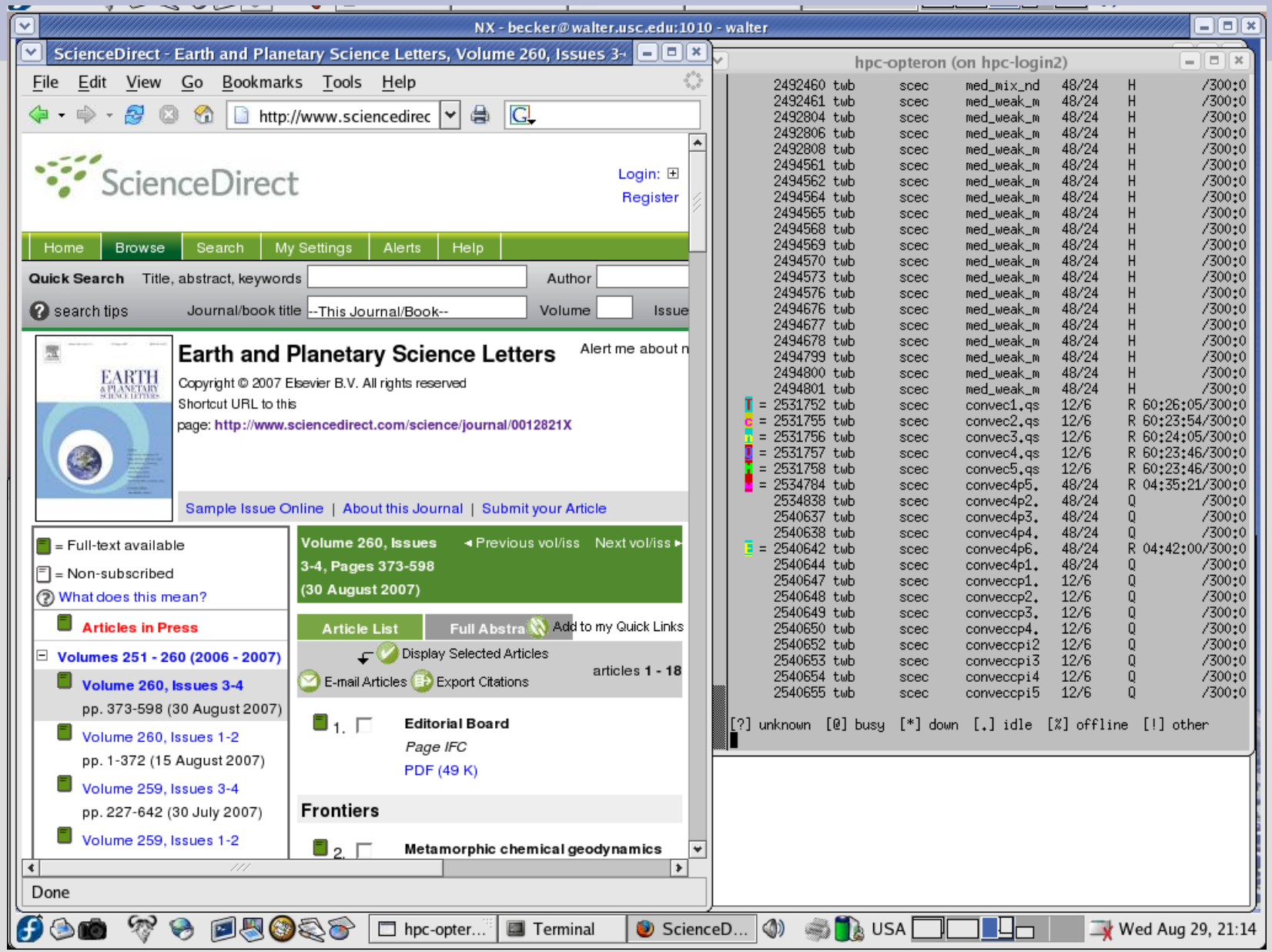
UNIX: Why use UNIX?

- can use same tools and programs on laptop, workstation, and supercomputer (less important if virtualization is available)
- flexible, modular, powerful
- seamless integration of C and F90 programs, shell commands, and post-processing (UNIX is written in C)
- all important numerical tools and libraries are available
- LINUX is open (security!), and ubiquitous

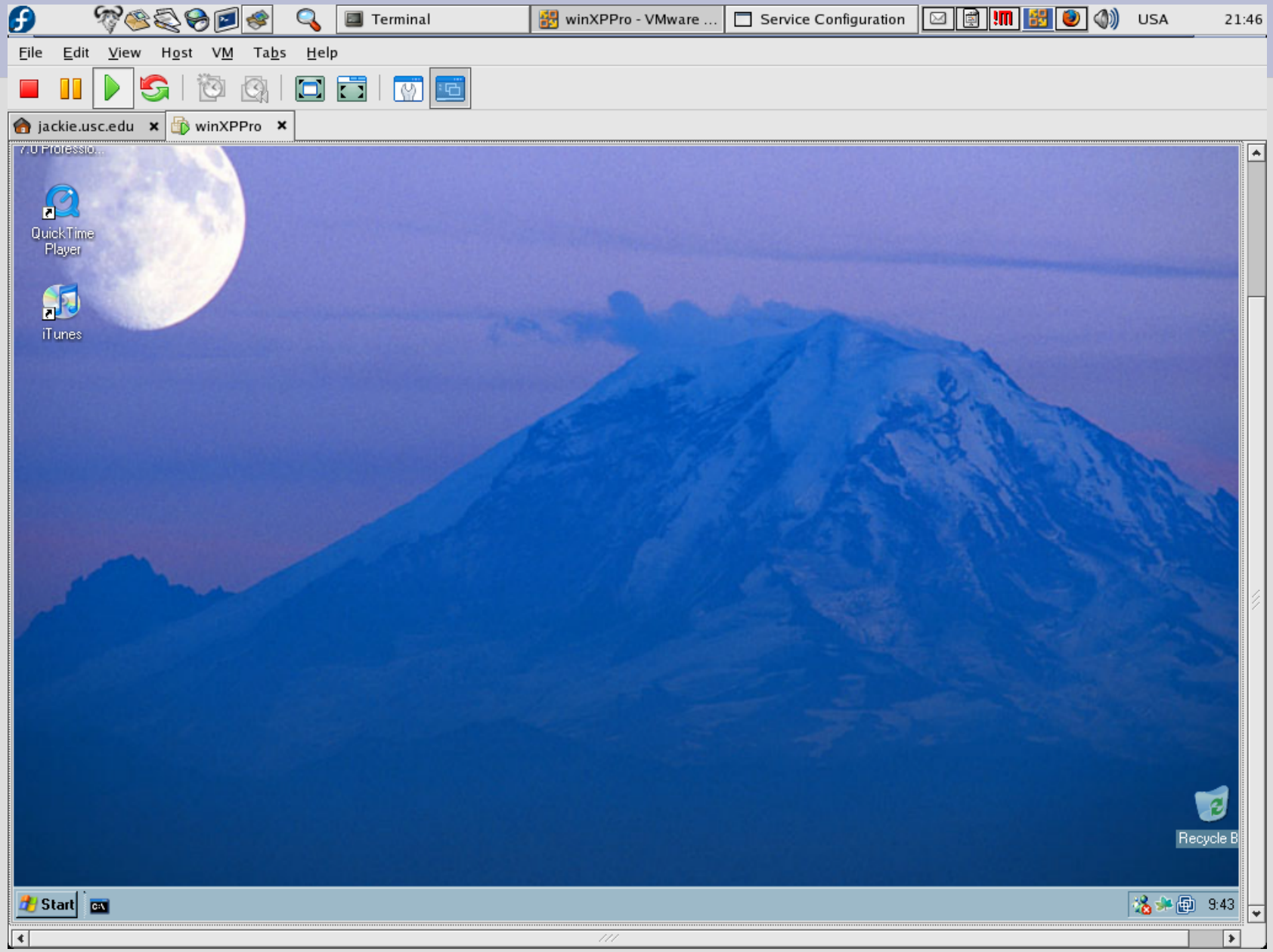
UNIX: Why (not) use UNIX?



Remote desktop: NX export your workplace everywhere



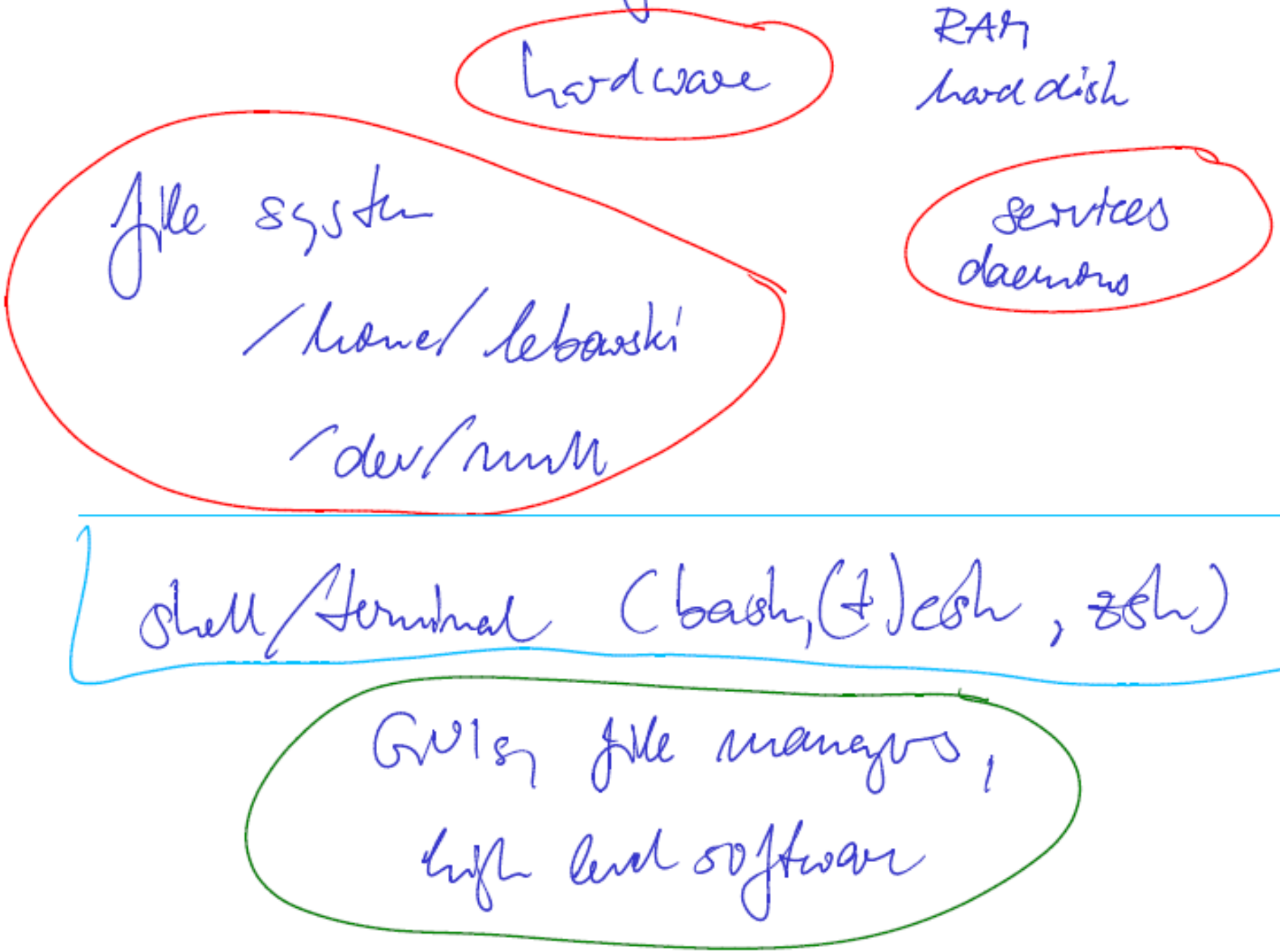
Virtualization: run any OS on whatever



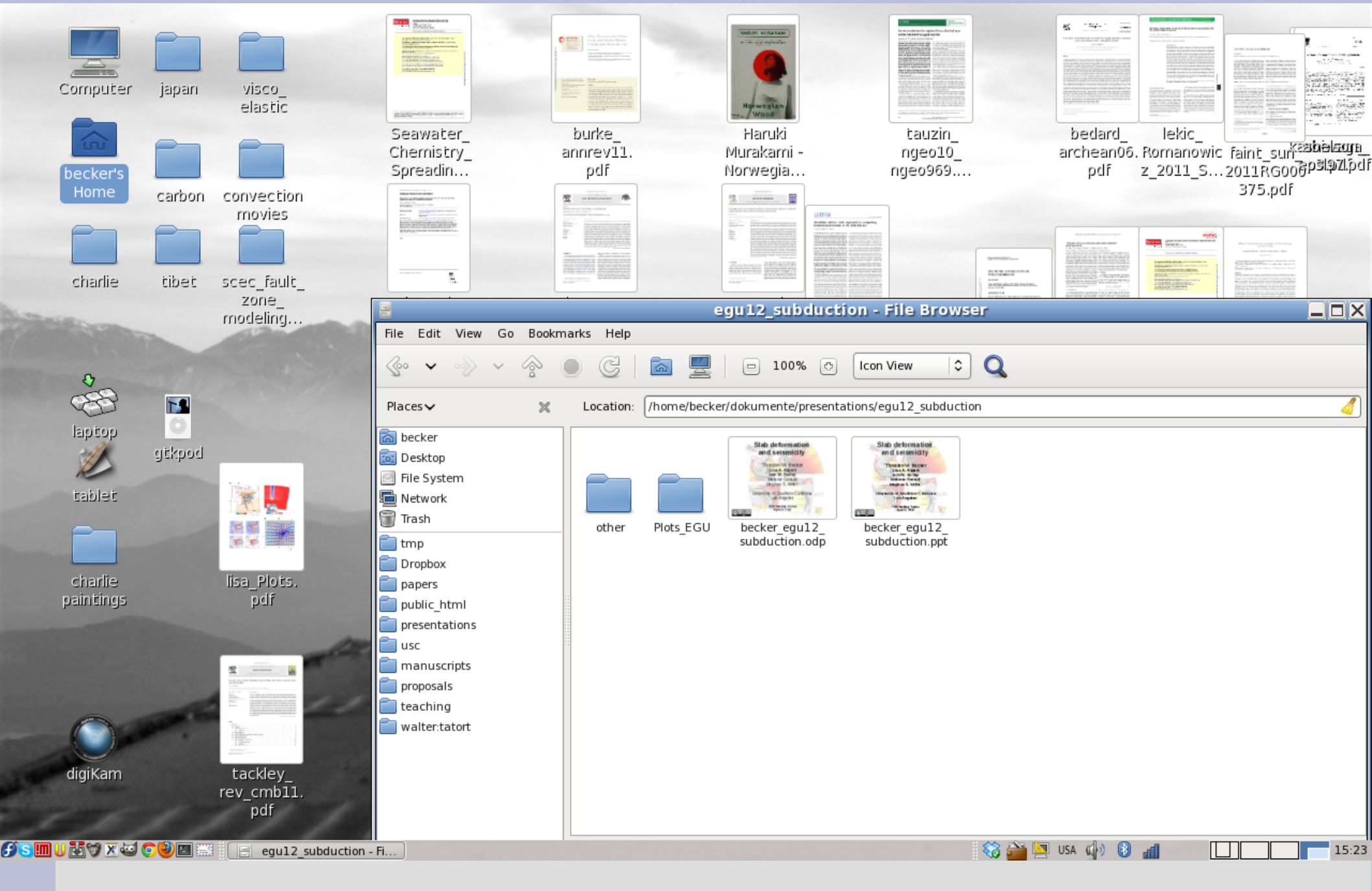
UNIX: This overview

- describes typical, ca. anno 2010, scientific workplace set-up in natural sciences
- tries to not
  - spend a lot of time on point-and-click GUIs
  - discuss vapor ware
  - discuss cutting-edge programs (with unclear support situation and user base)
- will be out of date tonight when it comes to specific software, but won't when it comes to UNIX and GMT
- Might be a bit old school (“old dog”)

Structure of a UNIX system

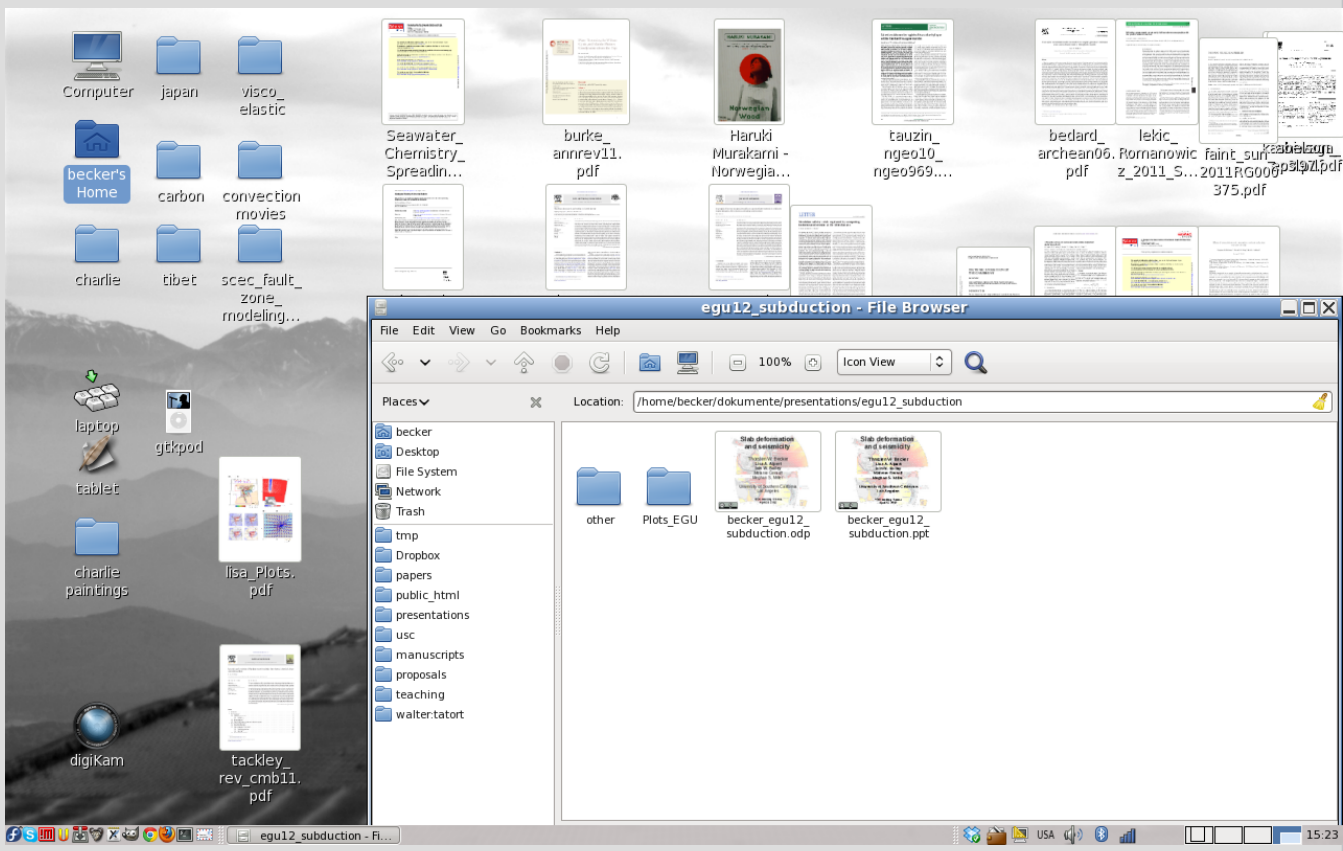


Hello: Gnome window manager



File system: Graphical Window managers and tools

- GNOME, KDE
- Provide support and interface with other apps (search, web, access files on other servers, etc.)



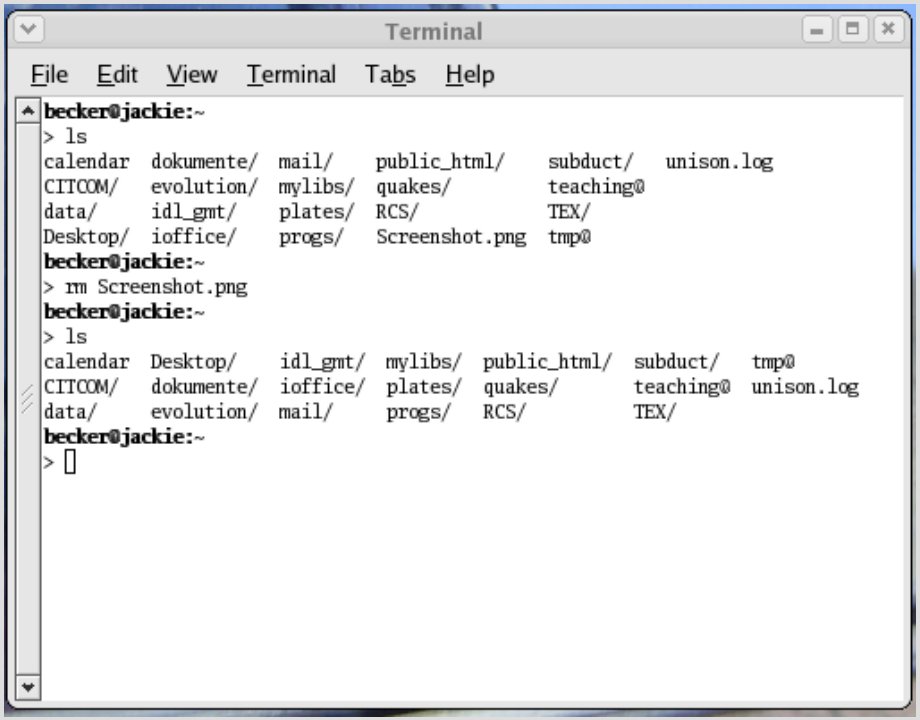
LINUX comes with a range of fairly useful GUI based software

- OpenOffice – for editing text (like Word), making presentations (like Powerpoint), and running spreadsheets (like Excel)
- GIMP – interactive image manipulation
- The only real problem is Illustrator.
- My suggestions for a single platform:
  - OS-X user: you have Linux already
  - Windows user: run Linux in a virtual machine
  - Undecided: Use a Mac (if you can afford it)

File system:
But where is UNIX? The shell

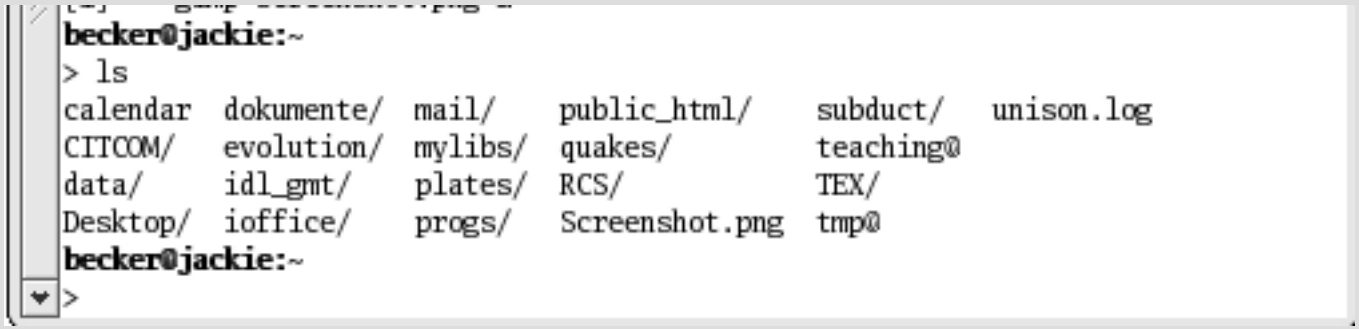
- open a *shell* (terminal) to get a *command line*
- type **commands**, such as **ls** to list the contents of a directory

Even if you regularly use Mac OS-X or GNOME, some knowledge of the background can save the day!

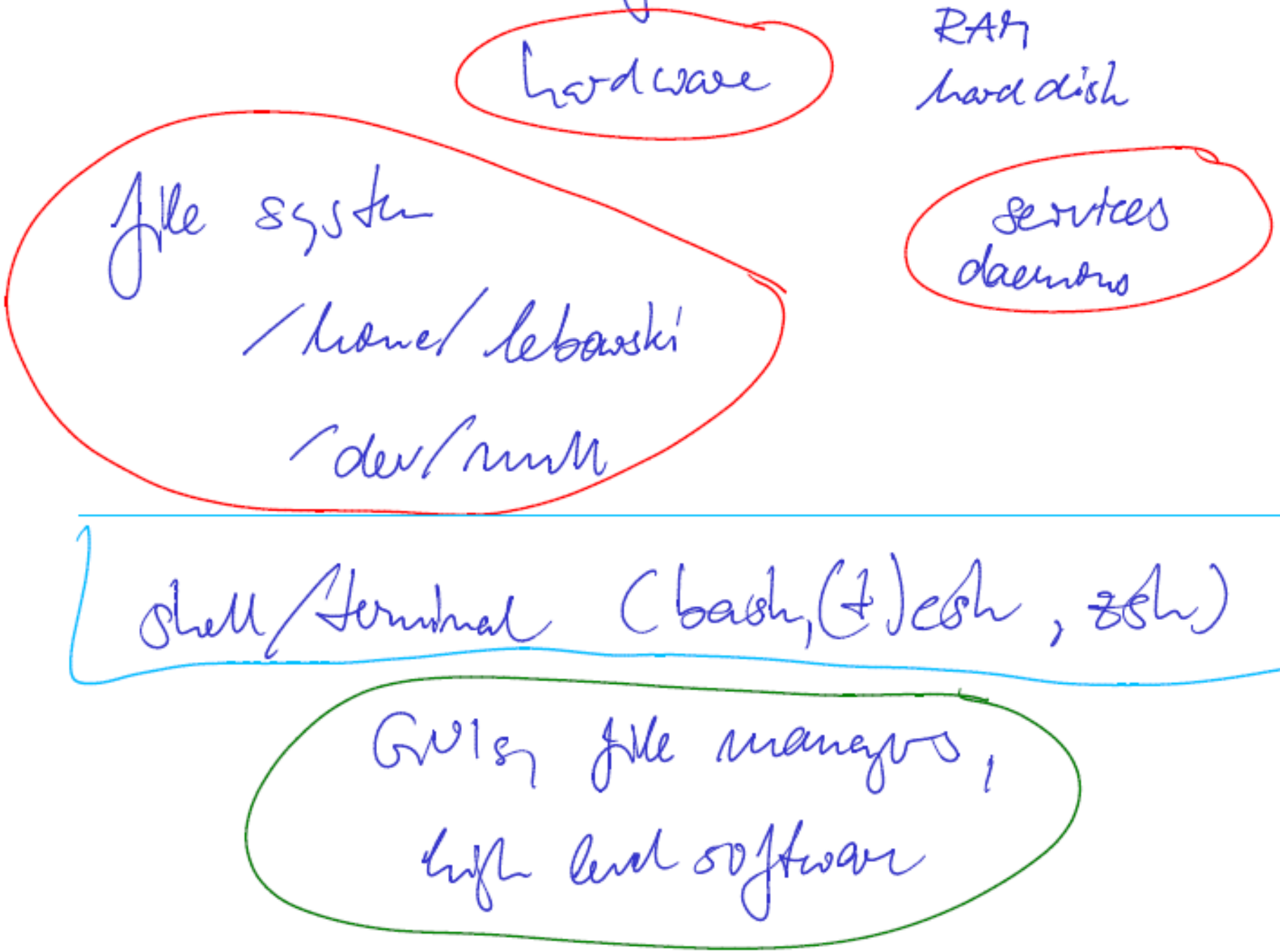


File system: Hardcore: The actual file system

- user *versus* super-user (administrator) setup
- tree structure of files within directories:
  - /usr/local has software
  - /dev has devices
  - /home/$USER has all the user's files, which might be subdivided into folders like
  -
  -
  -
  -
  -
  -
  - /mnt/data/ might hold shared data/storage



Structure of a UNIX system



Experimentation time
(q, CTRL-C to stop, CTRL-Z to pause)

- echo hello world
- passwd
- date
- hostname
- arch
- uname -a
- uptime
- who am i
- who
- id
- last
- w
- time sleep 5
- history
- last
- w
- top
- echo $SHELL
- echo {con,pre}{sent,fer}{s,ed}
- ls
- man ls
- clear
- cal 2012
- top
- locate passwd
- df, df -h
- du -sh .

File system: Naming conventions

- *suffixes* indicate type of file: file.dat, file.c, file.f, file.f90, file.awk, file.txt, file.tex, file.ps (and determines helper applications)
- UNIX is case sensitive
- normally, use lower case for files and directories
- some symbols (*e.g.*: *, %, ?) are special, if you want those literally you got to quote (*, \%, \?)
- different quotes (”, ', `) have different meanings

File system:

ls: list contents of directories

```
becker@jackie:~ > ls
calendar  data     dokumente  idl_gmt  mail    plates  public_html  RCS             subduct   TEX
unison.log
CITCOM    Desktop  evolution  ioffice  mylibs  progs   quakes       Screenshot.png  teaching  tmp
becker@jackie:~ > ls -F -l
total 6500
-rw-r--r--   1 becker users    1638 Jun 17 07:39 calendar
drwxrwxr-x   4 becker users    4096 Jun 17 07:39 CITCOM/
drwxr-xr-x  35 becker users    4096 Jul 12 15:22 data/
drwx------   2 becker users    4096 Jul 26 17:20 Desktop/
drwxr-xr-x  25 becker users    4096 Jul 12 07:48 dokumente/
drwx------   7 becker users    4096 Jun 17 11:53 evolution/
drwxr-xr-x   3 becker users   20480 Jul 27 15:00 idl_gmt/
drwxr-xr-x   3 becker users    4096 Jun 17 07:39 ioffice/
drwx------   2 becker users    4096 Jul  7 12:21 mail/
drwxr-xr-x  15 becker users    4096 Jun 17 07:46 mylibs/
drwxr-xr-x  12 becker users    4096 Jun 17 07:46 plates/
drwxr-xr-x  12 becker users    4096 Jun 17 07:46 progs/
drwxr-xr-x  27 becker users   12288 Jul 18 19:20 public_html/
drwxr-xr-x   4 becker users    4096 Jun 17 07:47 quakes/
drwxrwxr-x   2 becker users    4096 Jun 17 07:39 RCS/
-rw-r--r--   1 becker users   35775 Jul 27 16:15 Screenshot.png
drwxrwxr-x   5 becker users    4096 Jun 17 07:47 subduct/
lrwxrwxrwx   1 becker users      19 Jun 17 07:39 teaching -> dokumente/teaching//
drwxr-xr-x  29 becker users    4096 Jul 26 17:39 TEX/
lrwxrwxrwx   1 becker users      12 Jun 16 17:28 tmp -> /mnt/dos/tmp/
-rw-------   1 becker users 6508582 Jul 27 15:01 unison.log
```

File system: Permissions

```
-rw-r--r--   1 becker users    1638 Jun 17 07:39 calendar
```

u g a — user group size ctime filename

- first character: - (file), d (directory), l (link)
- r: read w: write x: execute or list
- u: user g: group a: all o: other
  - **chmod** u+x file
  - **chmod** a+r *.dat
  - **chmod** -R o-rwx my_stuff
- **whoami**, **id**: output of user and group

File system:
Commands have options

- command output and workings can be modified by adding *-x* (or x for **tar**)
- **ls**:
  - **ls** -F
  - **ls** -la
- usually, you can do "**command** --help" to learn more
- often, there are long version: **ls** --all --full
- man pages (RTFM): "**man** command"

File system:

If you like some options

- Use an alias

```
#alias convert '/usr/bin/convert -density 150 -background white -flatten -trim +repage'
unalias rm
alias opteron 'ssh -X -f twb@hpc-opteron.usc.edu xterm -ls -title hpc-opteron'

alias bunzip 'bunzip2'

alias epsmerge 'epsmerge --paper letter -par --print --postscript -lmar 0.01 -rmar 0.01 -tmar 0.01 -bmar 0.01 
-xcs 0.1 -ycs 0.1'

alias mroe more
alias cd.. 'cd ..'

alias ls 'ls -F'

alias new 'ls -ltF| more'

alias m 'more'

alias t 'tail'

alias h 'head'
```

File system:

File system commands I

- **cp**: copy files (will normally overwrite!)
  - **cp** filea fileb
- **rm**: remove files (for real!)
  - **rm** goneforever.dat
  - **rm** -i goneforever.dat
- **mkdir**: make directories
  - **mkdir** new_dir/
- **cd**: change directories (**cd** ..; **cd** -; **cd** ~)
- **pwd**: print current directory

File system:
File system commands II

- **scp**: copy files across machines
  - **scp** filea user@machine.usc.edu:~/directory/fileb
- **more**: display files page by page
  - **more** filea.dat
- **ln**: create (symbolic) links (*shortcuts* in Windows)
  - **cd** new_dir
  - **ln** -s ../old_dir/script .
  - soft *vs.* hard: deletion of hard link deletes file

UNIX tools: Command line tools for file management

- **more, less**: display files page by page interactively
- **cat**: display file
- **head**: display first few lines of file
- **tail**: guess
- **paste**: align files with columns row by row
    - **paste** file1.dat file2.dat
- **wc**: count words, lines, and bytes of file

Combining tools: Pipes and redirection (bash example)

- **>**: redirect stdout, **<**: stdin, **2>**: stderr
- **>>**: append, **|**: pipe
  - **cat** file1.dat > combined.dat
  - **cat** file2.dat >> combined.dat
  - **cat** file1.dat | **wc**
- myconvectioncode.exe < input.dat
- **echo** Whatever! > /dev/null
- mycode > log.dat 2> error.dat

UNIX tools: grep and sort

- **grep**: find patterns in file
  - **grep** my_function *.c | **more**
  - **grep** -ni my_function.*c (disregard case and list line numbers)
- **diff**: compare content of files (see **meld**)
  - **diff** file1.txt file2.txt
- **sort**: sort row data
  - **sort** -n +2 file.dat
- **uniq**: only print unique lines
  - **sort** -n splitting.dat | **uniq** > stations.dat

File system:
Using regular expressions

- * (all): **cp** *.dat new_dir/
- [pat] (pattern): **cp** file[1-5].dat new_dir
- ? (single letter/number): **cp** file??.dat new_dir
- **rm** -rf * (*DON'T TRY IT, IT WORKS*)